Machine Learning for the Booster Gradient Magnet Power Supply

Jason St. John for the GMPS-AI team
Accelerator Division

AI for Accelerators Workshop - 2022.01.14

Outline

- Challenge:
    - High-Precision Regulation for the Booster Gradient Magnet Power Supply
- Machine Learning Approach:
    - Data selection & Harvesting Infrastructure
    - Digital Twin: generative LSTM
    - Twin as Environment: Reinforcement Learning for a simple MLP
    - Deployment
        - hls4ml & FPGA bit-precision tests
        - Resource sharing & latency
        - Future steps
- Status

References herein as drawn from pre-print.

Fermilab

Support & Teamwork

Javier Duarte

Lucy Huang
Malachi Schram

Rachael Keller

Christian Herwig, Diana Kafkes, William A. Pellico, Gabriel N. Perdue, Andres Quintero-Parra, Brian A. Schupbach, Kiyomi Seiya, & Nhan Tran

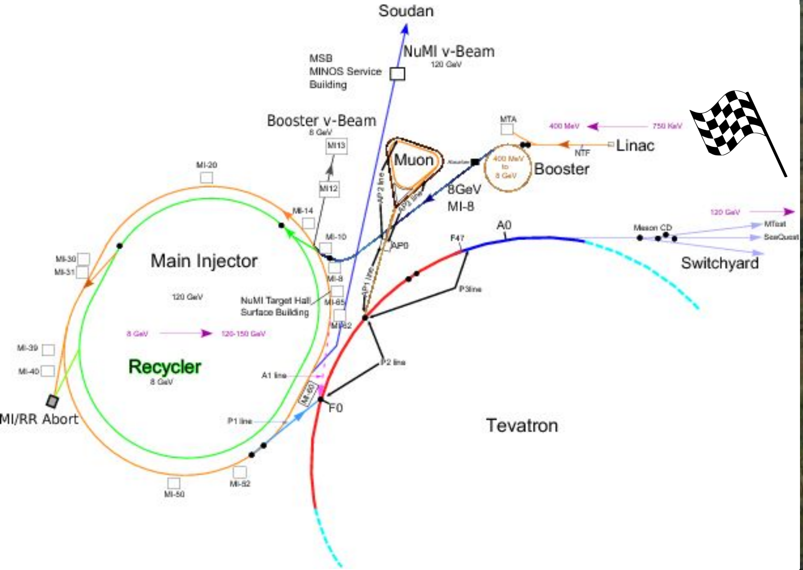
Gratitude for the use of resources of Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy (DOE), Office of Science, HEP User Facility, especially Fermilab Laboratory Directed Research and Development Program, Project ID FNAL-LDRD-2019-027.

Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359

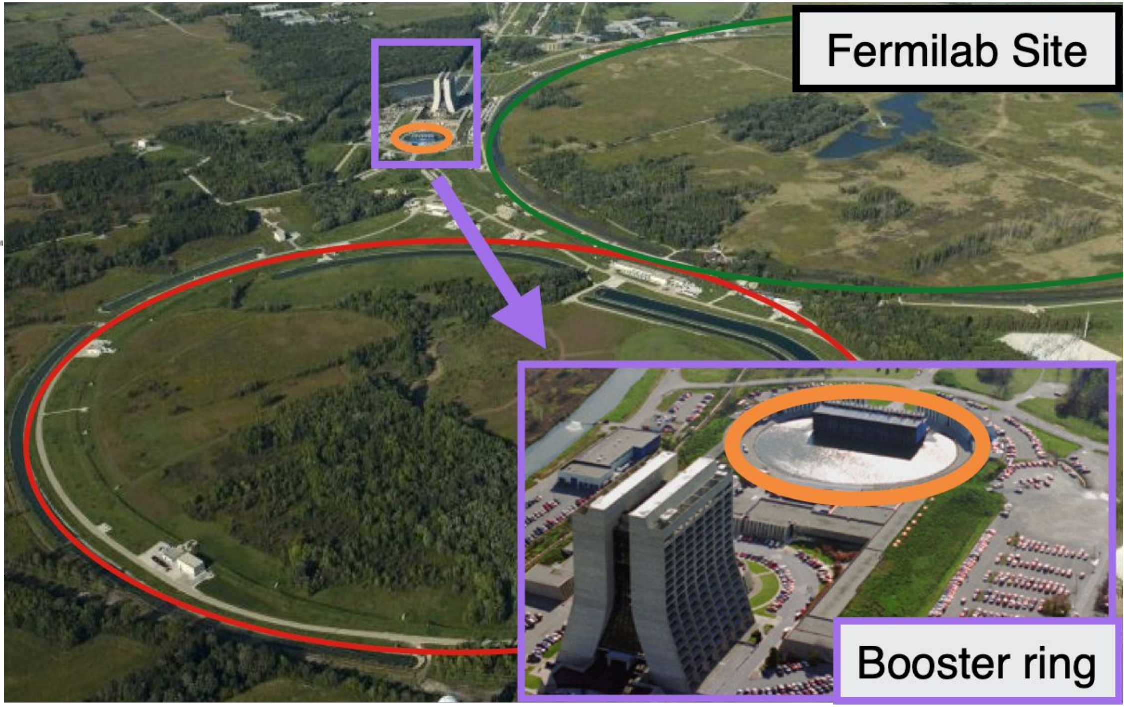
Pacic Northwest National Laboratory (PNNL) is a multi-program national laboratory operated by Battelle for DOE under Contract No. DE-AC05-76RL01830. Support here from DOE office HEP.

GMPS: Gradient Magnet Power Supply in Booster

Main bending & focusing magnets of Booster (smallest ring), which *boosts* 0.4 GeV H^- from Linac ⇒ 8 GeV H^+ for a wide array of High Energy Physics

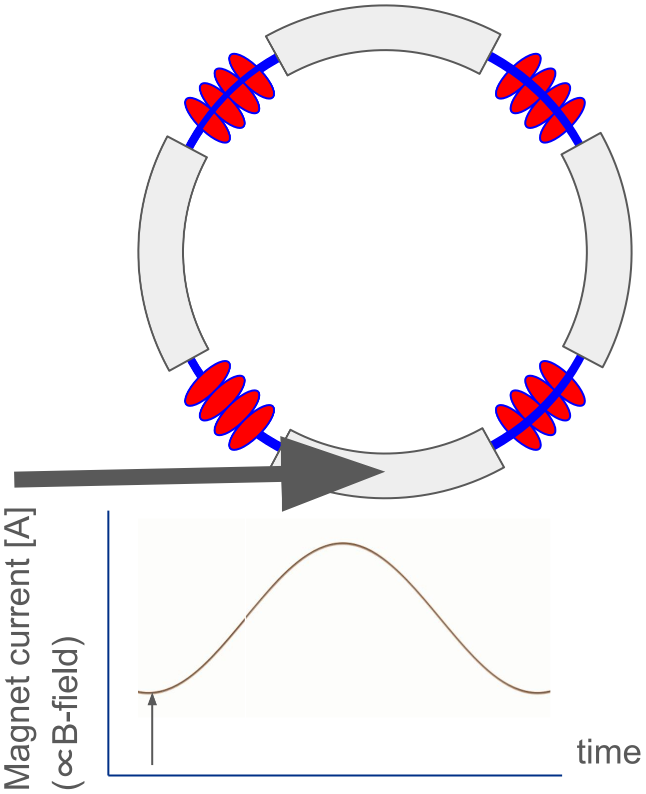


Negative hydrogen ions begin at the checkered flag and flow through the complex in pulses.

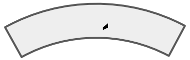


Fermilab

GMPS: Gradient Magnet Power Supply



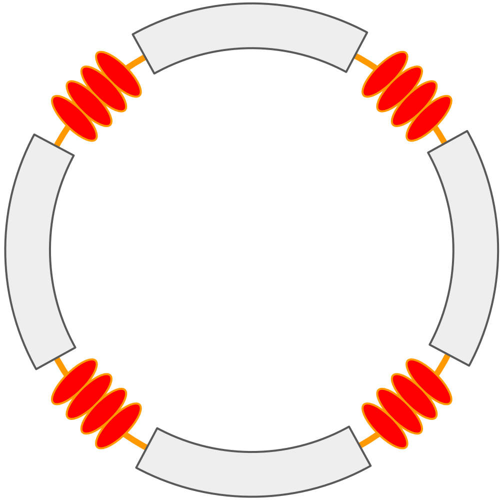
Inject at ~400 MeV/c from Linac

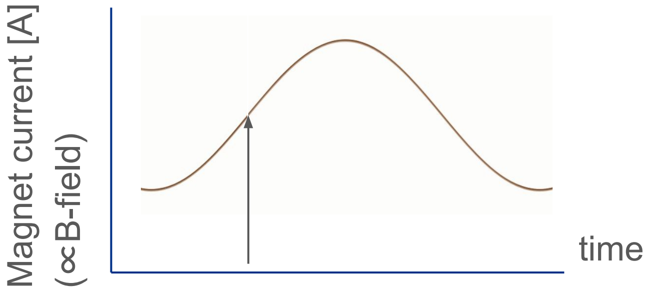
Bending magnets  need ~102 A to keep 400 MeV beam on orbit

H^- electrons stripped by passing through foil upon injection → making them H^+

Fermilab

GMPS: Gradient Magnet Power Supply





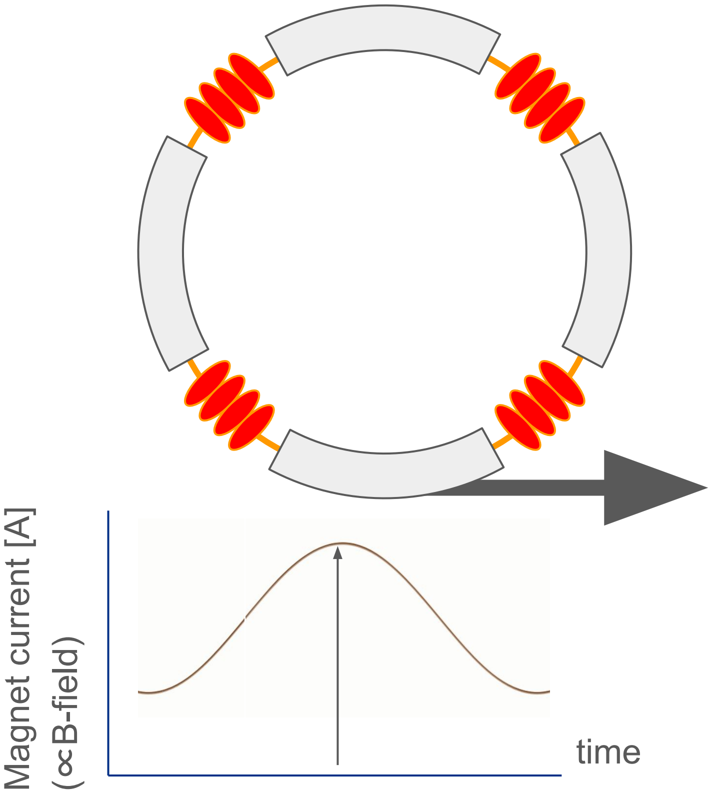
RF (Radio frequency) accelerator cavities capture beam and accelerate beam from

0.4 GeV/c → 8.0 GeV/c

following bending magnets in synchrony.

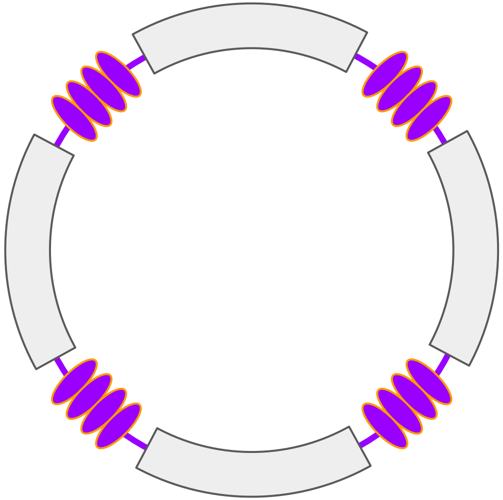
RF control is outside the scope of this project.

GMPS: Gradient Magnet Power Supply



Extract beam at maximum energy.

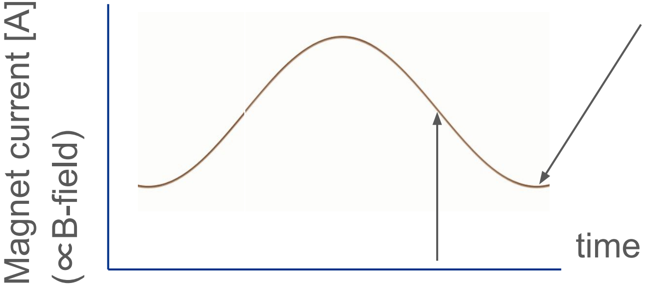
GMPS: Gradient Magnet Power Supply



Ramp back down to prepare for the next injection.

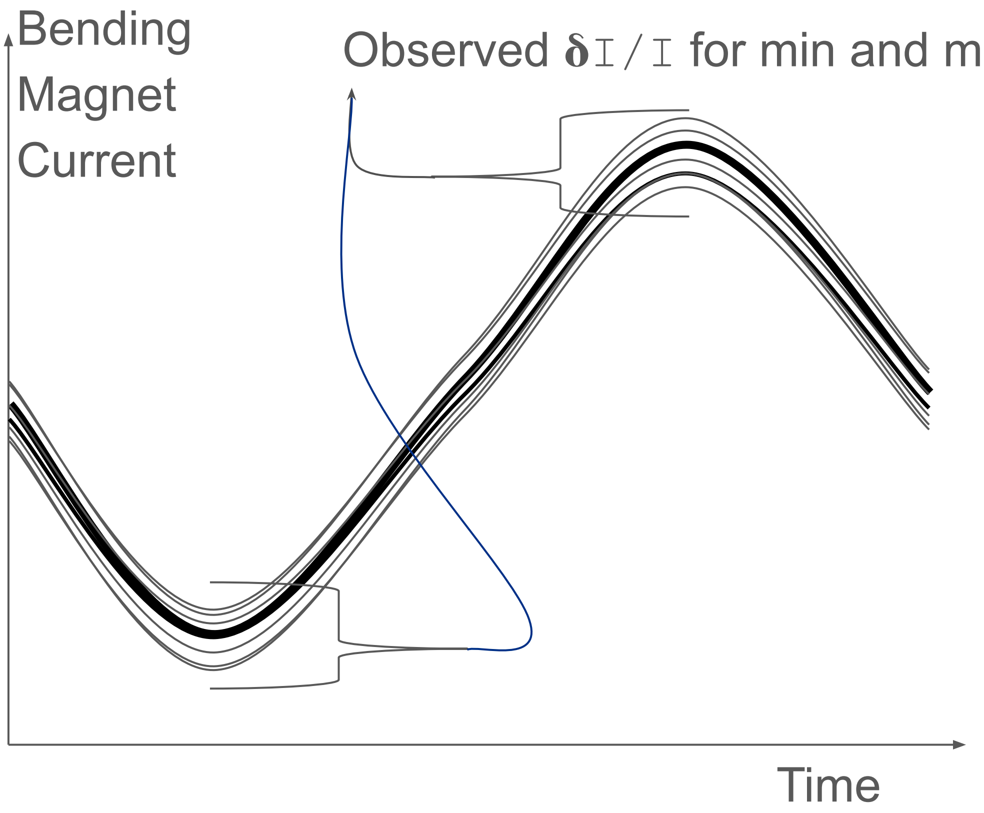
Complete cycle executes at 15 Hz

Resonant RLC circuit ⇒ Nonlinear response



Next cycle begins at
t = 1/15 s

GMPS AI: The Need for Improving Regulation



Perturbing influences:

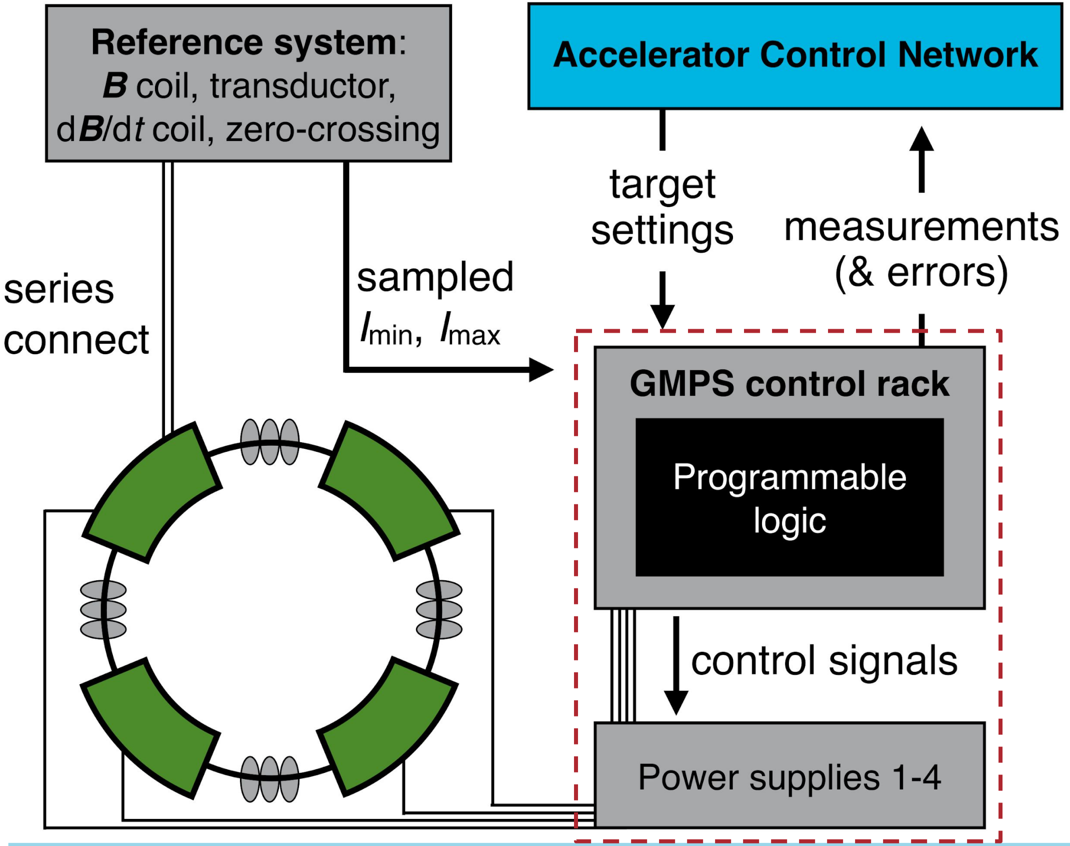
- Recent corrections made
- Other nearby synchrotrons
- Fluctuation of 60 Hz power
- Temperatures, etc

Available data mostly with the current PID regulator

Spread in B-field degrades beam quality, degrades repeatability, & contributes to losses

Fermilab

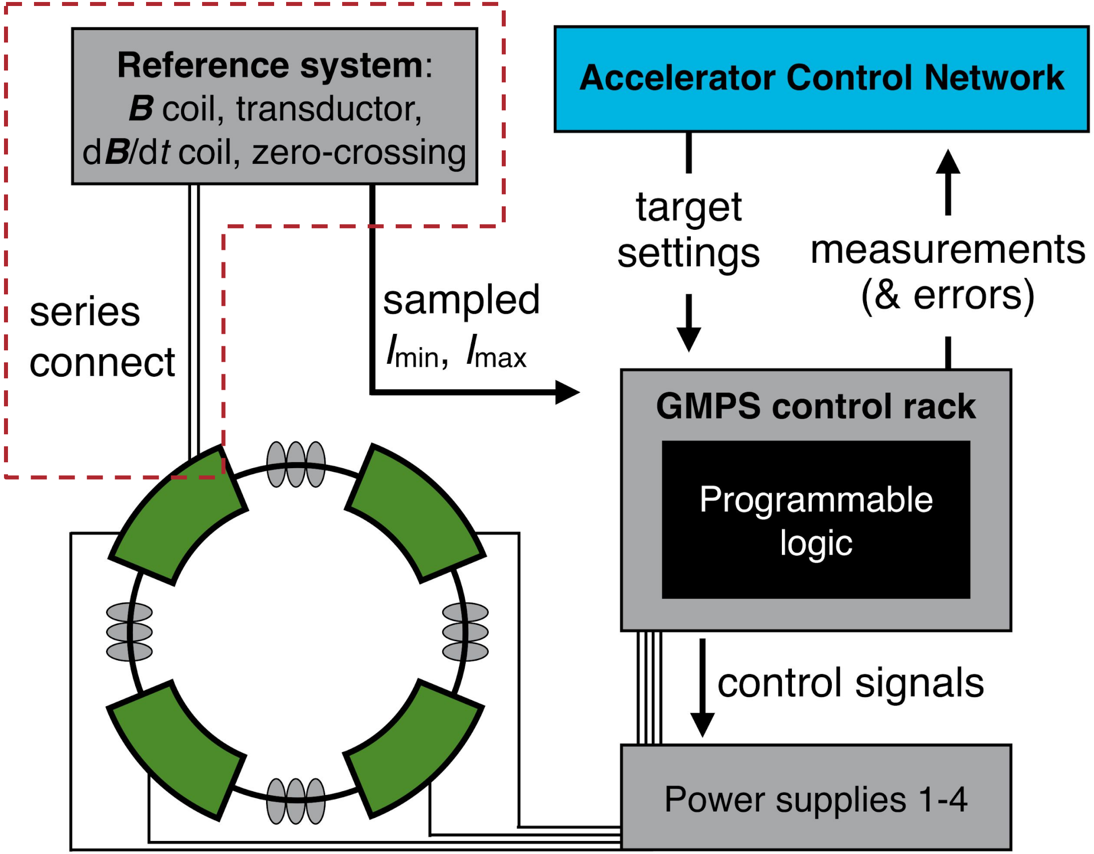
GMPS AI: Existing PID Circuit Regulation



GMPS PLC sends control signals to 4 bending magnet power supplies

Fermilab

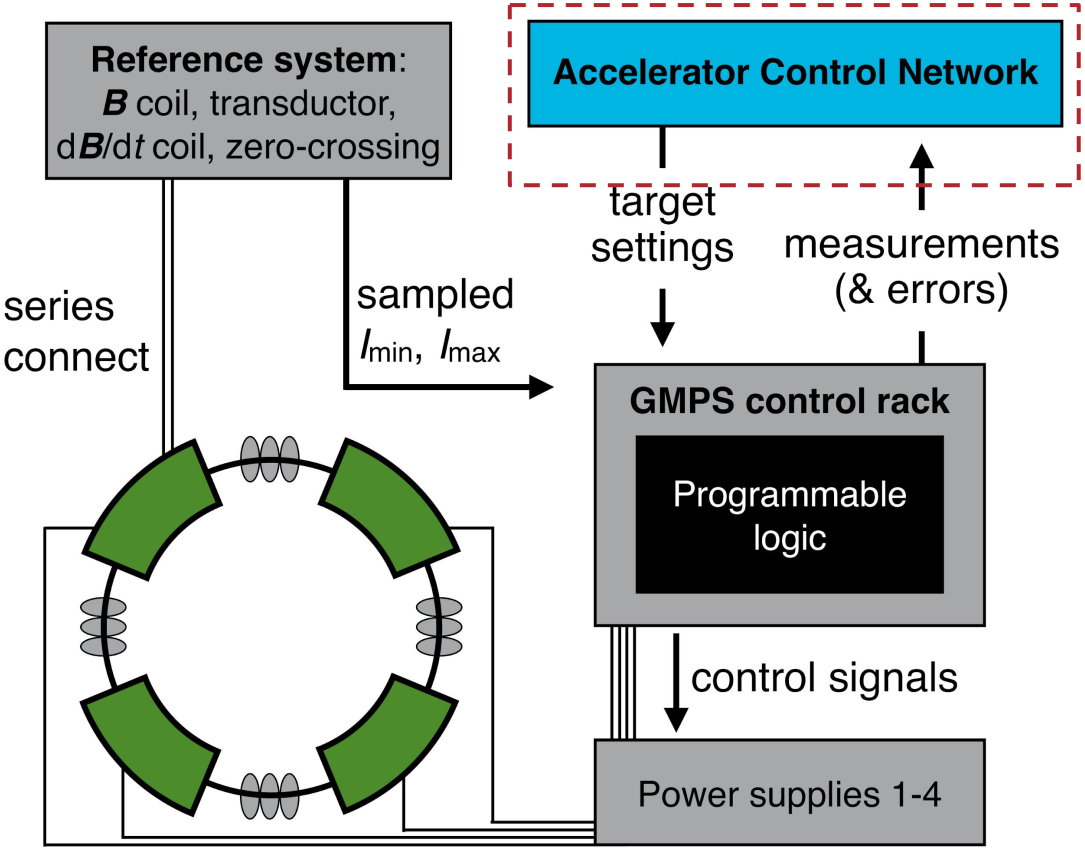
GMPS AI: Existing PID Circuit Regulation



Reference magnet with
B-field transductor

→ Zero-crossing time
→ Minimum field
→ Maximum field

GMPS AI: Existing PID Circuit Regulation

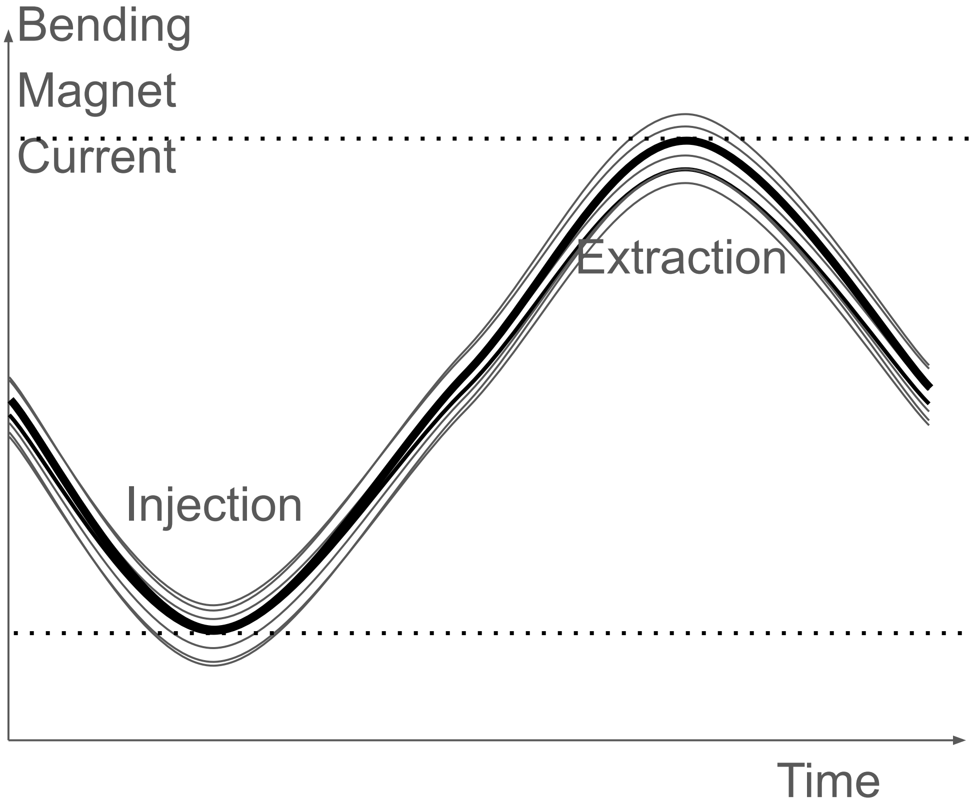


Human experts adjust target settings from time to time via control system

Also records settings & readings with some unknown latency

Known factors excluded from PID control logic:
Line Voltage variation,
Gallery temperatures, etc.

GMPS AI: Available time series data

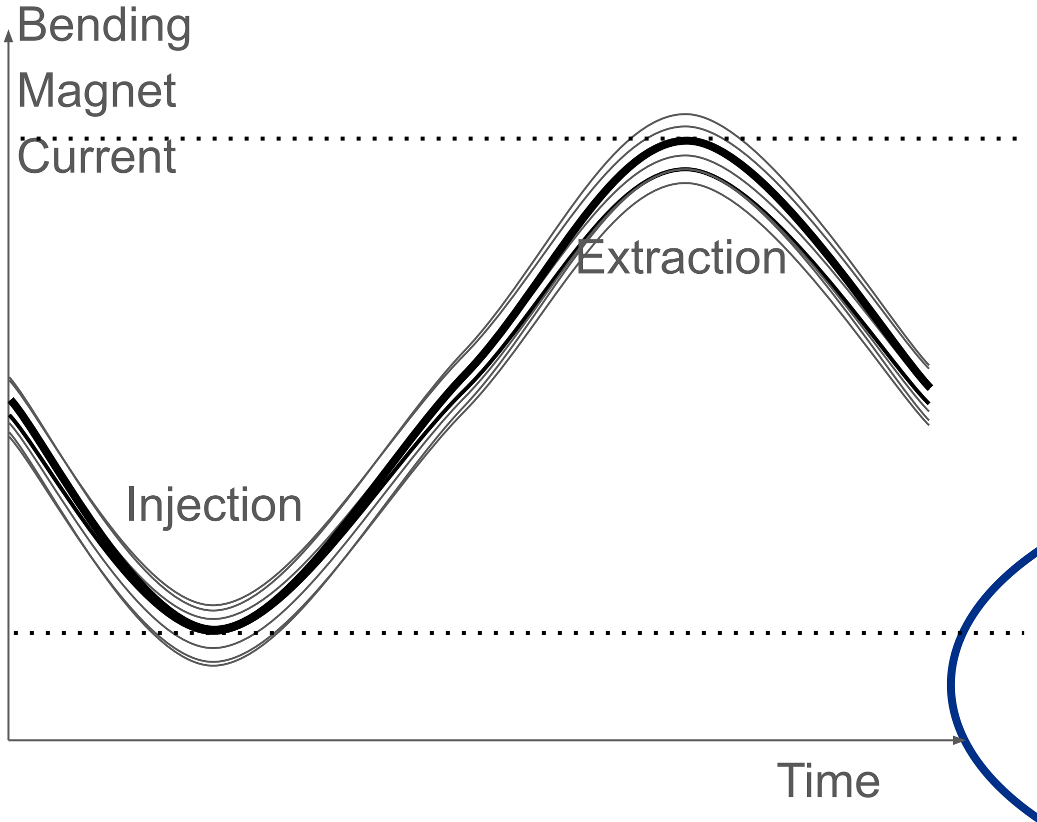


B_VIMAX = Setting to achieve
B:VIMAX = Prescribed remedy from PID regulator circuit
B:IMAXER = 10*(Setting - obsMax)

Independent regulation problems

B_VIMIN = Setting to achieve
B:VIMIN = Prescribed remedy from PID regulator circuit
B:IMINER = 10*(Setting - obsMax)

GMPS AI: Available time series data



We chose to focus on injection

- Simplifies development
- Can generalize once performing well
- Greatest potential for positive impact on science program

B_VIMIN = Setting to achieve
B:VIMIN = Prescribed remedy from PID regulator circuit
B:IMINER = 10*(Setting - obsMax)

GMPS AI: Selected time series data & dataset cleaning

Initial selection by Subject Matter Experts: 54 time series (out of 200k+ devices)
For small time window with constant settings, further narrowed to these five.
→ Biggest perturbation from MI current **I:MDAT40**
(Confirmed by Granger causality study vs “loss” **B:IMINER**)

Post-processed data: At every cycle, take most recent value for each device.
(Handles asynchronous timestamps.)

B:LINFRQ = 60 Hz line frequency error [mHz]
I:IB = MI lower bend current [A]
I:MDAT40 = MDAT measured MI current [A]

B_VIMIN = Setting to achieve*
B:VIMIN = Prescribed remedy from PID regulator circuit
B:IMINER = 10*(Setting - obsMax)

Fermilab

GMPS AI: Generative Multivariate LSTM as Digital Twin

Trained an LSTM to accurately predict next time step.

In “Ouroboros” configuration, this reproduces the learned multivariate dynamics, providing an offline environment to train a control agent through Reinforcement Learning.

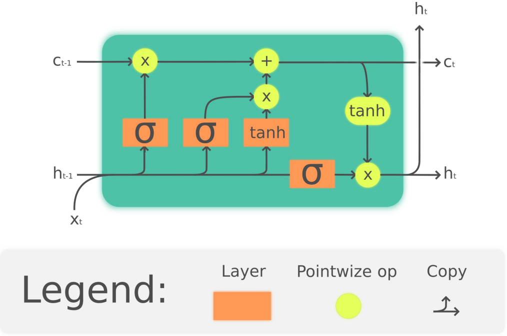
Last 150 timesteps

B:IMINER
B:LINFRQ
B:VIMIN
I:IB
I:MDAT40

B:IMINER
B:LINFRQ
B:VIMIN

Next Timestep

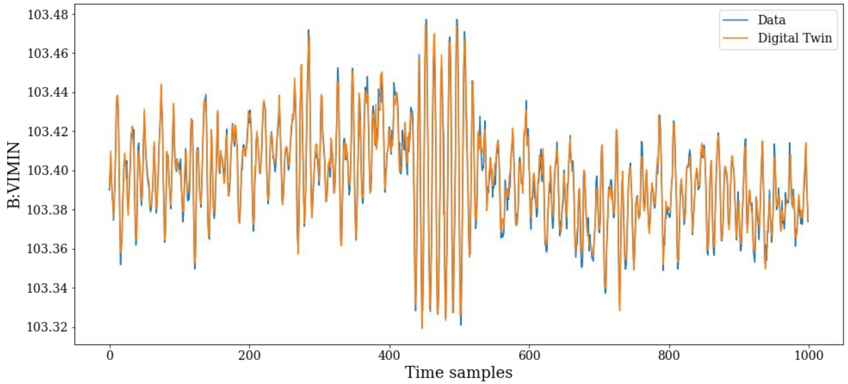
Agent adjusts **B:VIMIN**
(**I:IB** & **I:MDAT40** continue to draw from data)



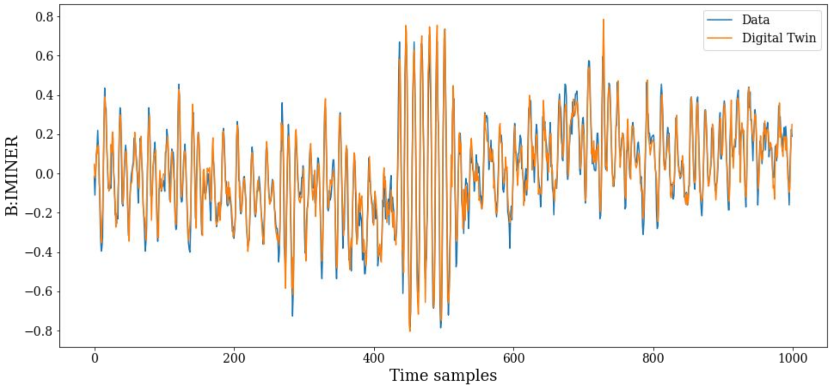
Long Short-Term Memory: A family of Recurrent Neural Network architectures featuring an hidden state, giving ability to learn long-timescale behaviors from data

GMPS AI: Generative Multivariate LSTM as Digital Twin

Results reflect behavior in data remarkably well.

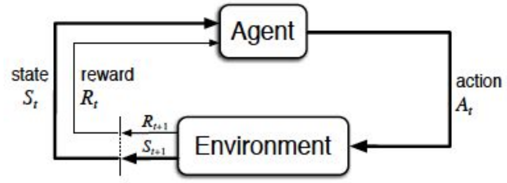


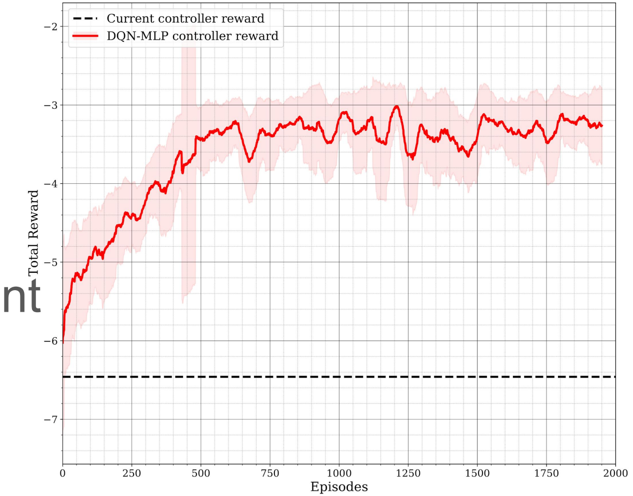
B:VIMIN = Prescribed remedy from PID regulator circuit



B:IMINER = 10*(Setting - obsMax)

GMPS AI: Digital Twin as RL Environment



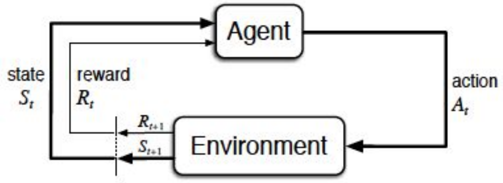


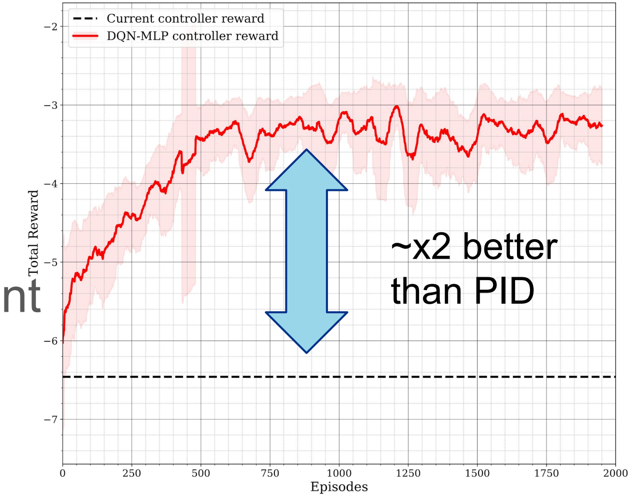
With LSTM providing environment, trained an MLP agent to tweak **B:VIMIN** prescription each timestep

- Reward function: neg. abs. error = -|**B:IMINER**|
- Q-learning @ 50 timestep episodes
  - Double DQN (target & policy model distinct)
  - 32-experience (random) to update policy model
  - ϵ-greedy decay factor 0.9995 (min: 0:0025)
  - Discretized options to change **B:VIMIN**:
    0 (no change), ±0.0001, ±0.005, and ±0.001.
  - 3 layers of 56 ReLU nodes

Fermilab

GMPS AI: Digital Twin as RL Environment





With LSTM providing environment, trained an MLP agent to tweak **B:VIMIN** prescription each timestep

- Reward function: neg. abs. error = -|**B:IMINER**|
- Q-learning @ 50 timestep episodes
    - Double DQN (target & policy model distinct)
    - 32-experience (random) to update policy model
    - ϵ-greedy decay factor 0.9995 (min: 0:0025)
    - Discretized options to change **B:VIMIN**:
      0 (no change), ±0.0001, ±0.005, and ±0.001.
    - 3 layers of 56 ReLU nodes

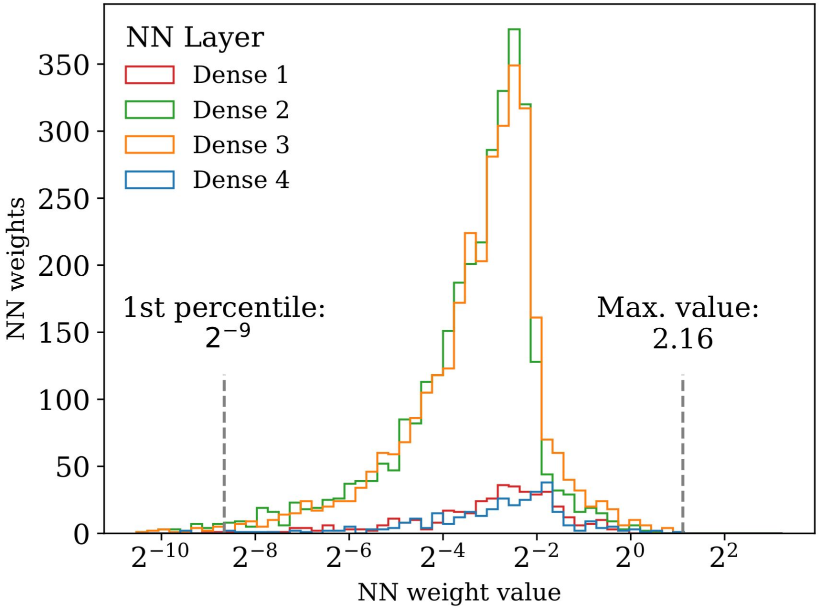
GMPS AI: Deployment on FPGA: Bit precision

7119 trained floats: How few bits can we use?

(Multiply-and-Accumulate) on 1518 available DSP slices

Layer	Outputs	Activation	Parameters	MACs
1	56	ReLU	336	280
2	56	ReLU	3192	3136
3	56	ReLU	3192	3136
4	7	Linear	399	392
Total	...	...	7119	6944

- 99% of weights are $>2^{-9}$



GMPS AI: Deployment on FPGA: Bit precision

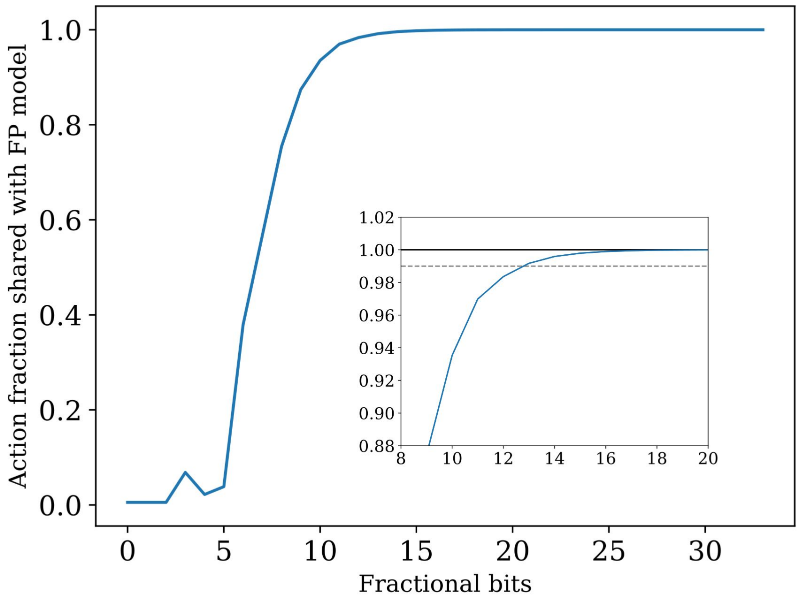
7119 trained floats: How few bits can we use?

(Multiply-and-Accumulate)
on 1518 available DSP slices

Layer	Outputs	Activation	Parameters	MACs
1	56	ReLU	336	280
2	56	ReLU	3192	3136
3	56	ReLU	3192	3136
4	7	Linear	399	392
Total	...	...	7119	6944

- 99% of weights are $>2^{-9}$
- >99.5% of actions are the same when using 14 bits to encode non-integer part of the weights

⇒ 1 (sign) + 5 (int) + 14 (fract.) = 20 bits



Fermilab

GMPS AI: Deployment on FPGA: Resources & Latency

Making it real: Keras model → Intel Arria 10

hls4ml to convert Keras models to High-Level Synthesis for FPGAs

DSP: Digital Signal Processor
(carries out MACs)
BRAM: Block RAM
MLAB: Memory Logic Array Block
ALM: Adaptive Logic Module
(simple arithmetic & logic operations)
Register: temporary value storage sites

reuse factor	DSP	BRAM	MLAB	ALM	Register	Latency
128	130	114	229	21.4 k	51.2 k	2.8 μs
224	74	100	1420	40.2 k	78.3 k	4.1 μs
1568	26	38	357	24.9 k	54.9 k	17.2 μs
Available	1518	2713	···	427 k	1.7 M	···

Reuse factor per layer = GCD of global reuse factor w.r.t. (inputs * outputs)

Tradeoff: Speed vs. Resource Usage Efficiency

GMPS AI: Status

- FPGA on dev board, talking to a server with GPU
  - First model loaded, replicates expected responses (Brian Schupbach)
  - Logging capabilities being added for dev board
    - Goal: Address data logger timestamp quality issue for AI@AD
- Found our online learning approach: Twin Delayed DDPG (T3D)
  - Control policy (neural net) running on chip, while an upgrade candidate is being developed. Gradual changeover.
  - Implies small change of neural net architecture (~~discrete~~ continuous), but retraining from scratch is fine
- Preparing for first real-time running
  - without settings (only log recommended actions)
  - then with settings. Expect engineering review.

GMPS AI: Future Steps

- Computing infrastructure for automated, continuous learning
  - Logging model parameters, performance, etc. also automatically, with hooks for human oversight
- Expanding dataset ~x1000 for LSTM
  - (Now the computing gets serious! ExaLearn.)
- Room for more sophisticated control agents. (So far <6% resource usage.)
  - Bigger MLP (offset with higher reuse factor?)
  - Parallel Ensembles voting
  - Data-driven model with Uncertainty Quantification (Environment & Agent)

Details about this project

Proof-of-concept pre-print paper aimed at Accelerator Physicists:

[2011.07371] Real-time Artificial Intelligence for Accelerator Control: A Study at the Fermilab Booster

Dataset used for these results, with ethical & technical Data Sheet:

BOOSTR: A Dataset for Accelerator Reinforcement Learning Control

Coming soon:

~x100 dataset on zenodo.org and in review with *Nature: Scientific Data*

Thank you!

GMPS AI: PID control logic

Based on history of current minimum error

$$B{:}IMINER_t = 10*(fitted\_min_t - B\_VIMIN_t)$$

generate cumulative time series (with ɣ=7.535008e-5)

$$\text{Beta: } \beta_t = \beta_{t-1} + \gamma B{:}IMINER_t$$

Now prescribe (with α=8.5e-2)

$$B{:}VIMIN_{t+1} = B\_VIMIN_t - \alpha\, B{:}IMINER_t - \beta_t$$

B_VIMIN = Setting to achieve

B:VIMIN = Prescribed remedy from PID regulator circuit

B:IMINER = 10*(Setting - obsMax)